

1 We claim:

2

3 1) A multi-thread processor for processing a plurality
4 of threads comprising:

5 a Thread ID generator producing a unique thread
6 indication for each said thread;

7 a plurality of register sets, one said register set for
8 each said thread, each said register set for each said
9 thread comprising a plurality of registers;

10 an n-way register set controller coupled to said
11 plurality of register sets and simultaneously handling
12 multiple read or write requests for one or more of said
13 unique threads;

14 a Fetch Address Stage for the generation of Program
15 Memory Addresses;

16 a Program Access Stage for receiving Program Memory
17 Data associated with said Program Memory Addresses;

18 a Decode Stage for converting said Program Memory Data
19 into instructions, said Decode Stage coupled to said n-way
20 register set controller;

21 a First Execution Stage for handling a multiply class
22 of instruction received from said Decode Stage;

23 A Second Execution Stage for handling an Arithmetic
24 Logical Unit class of instructions received from said Decode

1 Stage, said Second Execution Stage also coupled to said n-
2 way register set controller;
3 A Memory Access Stage for handling reading and writing
4 of external memory;
5 A Write Back Stage coupled to said n-way register set
6 controller for writing data to said register set;
7 each said Stage performing an operation during a Stage
8 Cycle;
9 said Thread ID value alternating from one stage cycle
10 to the next.

11

12 2) The processor of claim 1 where a pipeline core is
13 formed by stages in succession: said Fetch Address stage,
14 said Program Access stage, said Decode stage, said First
15 Execution stage, said Second Execution stage, said Memory
16 Access stage, and said Write Back stage.

17

18 3) The processor of claim 1 where said n-way register
19 set controller simultaneously receives at least one of read
20 requests from said Decode stage, read and write requests
21 from said Second Execution stage, or write requests from
22 said Write Back stage.

23

24 4) The processor of claim 1 where said Memory Access
25 stage is coupled to a memory controller.

1 5) The processor of claim 4 where said memory
2 controller issues a stall signal when receiving a memory
3 request to an external memory.

4

5 6) The processor of claim 4 where said memory
6 controller issues a stall signal when receiving a memory
7 read request to an external memory.

8

9 7) The processor of claim 4 where said memory
10 controller issues a stall signal which lasts an interval
11 from receiving a memory read request to receiving requested
12 data form said external memory.

13

14 8) The processor of claim 1 where said pipeline core
15 comprises a subset of said stages on one thread and
16 remaining said stages on said other thread.

17

18 9) The processor of claim 1 where said first execution
19 stage performs multiply operations and said second execution
20 stage performs non-multiply instructions.

21

22 10) The processor of claim 1 where said decode stage
23 forwards non-multiply operands to said second execution
24 stage.

25

1 11) The processor of claim 1 where program memory
2 contains a single instance of a program.

3

4 12) The processor of claim 1 where said thread ID can
5 be read by each said thread.

6

7 13) The processor of claim 1 where each said thread
8 reads said thread ID to perform thread operations which are
9 independent.

10

11 14) The processor of claim 1 where thread ID is used
12 along with an address to decode a device in a memory map.

13

14 15) The processor of claim 1 where devices are decoded
15 in a memory map based on address only.

16

17 16) The processor of claim 1 where said Decode stage
18 performs decoding of instructions for said multiply class of
19 instruction, and said First Execution stage performs
20 decoding of instructions for said arithmetic logical unit
21 class of instructions.

22

23 17) The processor of claim 16 where if one of said
24 multiply class of instructions requires a register operand,

1 said operand is provided from said registers to said decode
2 stage.

3

4 18) The processor of claim 16 where if one of said
5 arithmetic logical unit class of instructions requires a
6 register operand, said operand is provided from said
7 registers to said first execution stage.

8

9 19) The processor of claim 1 where at least one said
10 stage includes an operational clock which is at a higher
11 rate than said stage clock.

12

13 20) A multi-threaded processor comprising a plurality
14 of stages operating on a stage clock for passing information
15 from stage to stage, each stage including inter-stage
16 storage for thread information associated with a thread ID;

17 a first said stage receiving program counter address
18 information from a unique program counter for each said
19 thread ID and delivering said address to a program memory;

20 a second stage for receiving program data from a
21 program memory;

22 a third stage for performing decode of said program
23 data;

24 a fourth stage for performing multiplication operations
25 or decode operations;

1 a fifth stage for performing non-multiplication
2 operations;
3 a sixth stage for accessing external memory;
4 a seventh stage for writing results of computations
5 performed in said fourth stage or said fifth stage back to a
6 register set;
7 said register set being duplicated for each said thread
8 ID;
9 each said first through seventh stage receiving said
10 thread ID and operating according to a first or second
11 value.

12

13 21) The multi-threaded processor of claim 20 where said
14 first, third, fifth and seventh stages use one value for
15 said thread ID, and said second, fourth, and sixth stages
16 use a different value for said thread ID.

17 22) The multi-threaded processor of claim 20 where said
18 threads each control execution of a program, and said
19 programs execute independently of each other.

20

21 23) The multi-threaded processor of claim 22 where one
22 said thread may stop execution and the other said thread
23 continues execution.

24

1 24) The multi-threaded processor of claim 20 where said
2 registers and said stages contain data which is used
3 separately for each said thread ID.

4

5 25) The multi-threaded processor of claim 20 where said
6 stages alternate between two threads on each said stage
7 clock.

8

9 26) The multi-threaded processor of claim 20 where said
10 thread-ID identifies a register set and a program counter.

11

12 27) The multi-threaded processor of claim 20 where said
13 third stage performs said decode for multiply operations.

14

15 28) The multi-threaded processor of claim 20 where said
16 fourth stage performs said decode for non-multiply
17 operations.

18

19 29) The multi-threaded processor of claim 27 where said
20 fourth stage performs said multiply operations.

21

22 30) The multi-thread processor of claim 28 where said
23 fifth stage performs said non-multiply operations.

24

1 31) The multi-thread processor of claim 28 where said
2 non-multiply operations include at least one of rotate,
3 shift, add, subtract, or load.

4
5 32) The multi-thread processor of claim 29 where said
6 multiply operations include multiplication by a constant
7 from one of said registers.

8
9 33) The multi-thread processor of claim 30 where said
10 non-multiply operations include addition of a multiply
11 result from said fourth stage.

12
13 34) The multi-thread processor of claim 20 where said
14 thread ID includes a plurality of values, each said value
15 having at least one register and a program counter.

16
17 35) The multi-thread processor of claim 20 where said
18 sixth stage said external memory responds in more than one
19 said stage clock cycle.

20
21 36) The multi-thread processor of claim 20 where said
22 external memory generates a stall signal for each said
23 thread ID, thereby causing all said stages to store and
24 maintain data for that thread ID until said stall signal is
25 removed by said external memory.

1

2 37) The multi-thread processor of claim 20 where said
3 fifth stage generates an address for a data memory.

4

5 38) The multi-thread processor of claim 37 where said
6 sixth stage receives and generates data for said data
7 memory.

8

9 39) The multi-thread processor of claim 20 were said
10 thread information storage includes registers which store
11 results from said fifth stage for each said thread ID.

12

13 40) The multi-thread processor of claim 20 where said
14 registers which store results from said fifth stage allow a
15 non-stalled thread to continue execution without modifying
16 said stored results.

17

18

19

20